Decentralized AI: Meaningful utilization of computational power for real-world application

David Liberman
liberman@gmx.com

Abstract. This paper explores the development of a decentralized AI infrastructure designed to address the escalating computational demands of artificial intelligence. Despite the potential of decentralized systems, a significant challenge remains: much of the computational power in decentralized networks is currently dedicated to tasks that do not contribute to real-world utility. The aim is to construct a decentralized AI network that optimizes the use of hardware resources, aligning computational efforts with productive tasks. To achieve this, the "Proof-of-Work 2.0" mechanism is introduced as an advanced consensus protocol tailored for decentralized AI blockchain networks.

1. Introduction

Artificial Intelligence (AI) has become integral to various industries, driving innovation and transforming workflows across numerous domains. However, the increasing sophistication of AI systems has led to a dramatic rise in computational demands, particularly for key tasks such as training large models and executing inferences. Training involves optimizing model parameters by processing extensive datasets, requiring high-performance computing resources such as GPUs and TPUs. Inference, or applying trained models to new data, also demands significant computational power, particularly as models handle increasingly diverse and complex operations. These meaningful activities (AI model training and inference) are the primary focus of the proposed decentralized AI infrastructure, ensuring that computational resources are dedicated to productive outputs. Unlike many decentralized systems, where much of the computational power is misused on meaningless tasks, the novel decentralized AI network aims to handle these essential tasks efficiently, with minimal resource waste, making the infrastructure scalable and sustainable.

Many decentralized systems today suffer from a fundamental misalignment in resource allocation and incentives. A significant portion of rewards is directed toward network security rather than directly advancing meaningful AI computation, which wastes block subsidies. For example, in systems like Bittensor, 60% of rewards are allocated to staking, which, while necessary for network security, does not contribute to AI computation. Furthermore, the remaining 40% of rewards are also used inefficiently, as contributors to AI computation often repeat tasks to ensure accuracy and meet network validation requirements, leading to suboptimal hardware utilization. This misallocation of incentives reduces efficiency, diverting capital and computational power from meaningful AI advancements, thereby increasing costs.

The proposed decentralized AI network addresses these inefficiencies by ensuring that almost 100% of computational resources are directed toward meaningful tasks such as AI model training and inference. This is achieved through time-bound proof-of-work competition, as well as honest-majority-based validation, both described in detail below. This alignment of incentives with productive work eliminates subsidy wastage and optimizes the use of available hardware, providing a more efficient and cost-effective alternative to traditional infrastructures.

2. Centralization, decentralization, and main challenges

Centralization

The scalability and efficiency of AI infrastructure are crucial, as they directly impact AI systems' cost and performance. However, reliance on centralized infrastructure, primarily controlled by a few major cloud providers, introduces several critical challenges:

1. **High costs and monopolistic pricing.** Centralized cloud providers offer computational resources at a premium, structuring their pricing models to favor large-scale enterprises. [1][2][3] Smaller developers, lacking the bargaining power and economies of scale, are subjected to significantly higher costs. [4] This disparity is particularly evident when deploying advanced open-source AI models, such as Meta's Llama 3.1 405B, on leading cloud platforms, such as AWS or Azure. The expenses of running these models are comparable to those of proprietary solutions like OpenAI's API. [5] This parity in costs results from monopolistic pricing strategies employed by major providers. These providers maintain high margins for smaller clients while offering substantial discounts to larger entities [6][7], sometimes even operating at a loss to secure their business. As AI models continue to evolve, incorporating not just textual but also visual and auditory capabilities, the cost per user escalates significantly. Developers utilizing these advanced AI models face increasingly prohibitive expenses, particularly when relying on centralized cloud infrastructure. This is compounded by the fact that internet users, accustomed to free or low-cost services, are often unwilling to cover these rising costs, placing additional financial burdens on developers. Historically, the perceived affordability of open-source large language models (LLMs) was due to their smaller size. However, the associated costs have similarly increased as these models have expanded to match the parameter counts of proprietary models from OpenAI and Anthropic. Smaller cloud providers find it difficult to compete due to lower utilization rates, which prevents them from offering competitive pricing over the long term. This pricing dynamic forces many developers to consider purchasing hardware outright [8], which can reduce costs but sacrifice the flexibility needed to manage the variable demand generated by user requests effectively. Consequently, the current cost structures not only inhibit innovation but also restrict access to advanced AI capabilities, creating significant barriers for smaller entities in the AI ecosystem.
2. **Risks of censorship and centralized control.** Concentrating computational resources within a few dominant providers presents significant risks related to censorship and centralized control. [9][10] These centralized entities have the authority to impose restrictions, monitor usage, and potentially censor applications that do not align with their policies. [11][12] As AI systems continue to develop and disrupt various industries, this concentration of power becomes increasingly problematic. Instead of driving down costs and democratizing access to AI technologies across all sectors, the control exercised by a small number of players could lead to a scenario where the economic benefits of increased efficiency are captured by these few entities rather than being distributed more broadly. Such centralization threatens the openness and accessibility of AI technologies, stifling innovation by limiting the diversity of applications and models that can be developed and deployed. The potential for censorship and restricted access underlines the need for a decentralized AI infrastructure that not only addresses the issues of cost but also ensures that the benefits of AI are available to all without the risks associated with centralized control and censorship. By decentralizing AI resources, the technology can be made more equitable and accessible, fostering a future where AI provides widespread benefits and mitigates the risks of monopolistic control and undue influence.

Decentralization

Decentralization is often seen as a solution to the challenges of centralization, offering a way to distribute computational resources across a network of nodes of relatively small individual computational power, thereby eliminating the monopolistic control of centralized providers.

Modern decentralized networks also utilize additional incentive mechanisms, such as issuing new digital currency and providing a reward (known as block subsidy) with each transaction. These subsidies can effectively lower the cost of hardware and computational resources for the customer, especially during the initial stages of network development. In the context of AI, the subsidies have the potential to reduce the cost of training and inference for developers and customers by as much as 50% or more compared to current centralized providers, particularly during the network's initial growth phase.

The recent AI advancement brought up a new computational challenge that forced large and small players to invest billions in hardware. However, computational costs are expected to decrease substantially over time, both following Moore's law and because of the proven potential of crypto-subsidized networks to supercharge competition and breakthroughs in computational technology. As these costs decline, the need for continued subsidies diminishes, making the incentive structures of decentralized systems well-suited for supporting AI's evolving computational requirements.

In theory, decentralized systems should provide greater cost-efficiency, flexibility, and resilience. However, the architecture of modern decentralized systems exhibits significant inefficiencies. A large portion of the block subsidies and computational power is consumed on tasks that do not yield meaningful results, primarily focusing on network security rather than practical outputs. This inefficiency stems from two main factors: the high cost of achieving consensus and the high degree of redundancy built into a computation accuracy verification.

Substantial computational or capital resources are required to secure the blockchain through mechanisms like Proof-of-Work or Proof-of-Stake. These consensus models assume that most participants are honest but require all participants to allocate significant resources to prevent malicious minority actors from manipulating the network by making everyone believe they represent the majority. *The reasons for inefficiencies are described in more detail in Chapter 3.*

In addition to securing blockchain, in decentralized AI projects, the same computational tasks are executed multiple times to ensure the accuracy of the results. Similarly, decentralized storage projects may duplicate files multiple times for verification. This redundancy leads to considerable resource waste, particularly in AI, where even small task repetitions can drastically increase computational costs.[13] Moreover, even when results are verified by other nodes once or twice, there is no guarantee that these nodes are independent, and they may all be part of the same colluding group, thereby compounding the risk of false outcomes while maintaining the appearance of redundancy.

These inefficiencies also increase the environmental footprint, significantly raising energy consumption. The U.S. Energy Information Agency [14] estimates that cryptocurrency mining accounts for 0.6 to 2.3 percent of global electricity use annually, highlighting the high energy consumption associated with these processes. While meaningful AI tasks also require significant electricity, the inefficiencies caused by task repetition and consensus mechanisms result in a far greater waste of resources in current decentralized systems.

The need for decentralized AI infrastructure

Despite these challenges, there is a compelling need for decentralized AI infrastructure to overcome the limitations of centralized systems. Decentralized networks can foster a genuinely competitive market environment, unlike the oligopolistic nature of centralized cloud providers. In a decentralized ecosystem, numerous participants can contribute computational resources and benefit from the network's growth, promoting innovation and efficiency. Furthermore, issuing tokens or coins can enable significant upfront subsidies for hardware and software.

3. Traditional consensus mechanisms and their inefficiencies

Proof-of-Work

Bitcoin's security isn't only based on complex cryptographic algorithms. It fundamentally relies on the principle of honesty among its participants. Honest node, a computer running software, strictly follows predefined consensus rules, ensuring consistent and predictable behavior within the network. The Bitcoin white paper underscores the importance of this by mentioning "node" 38 times and "honest" 16 times, often together, to ephasize the trustworthiness of those operating the network. On the flip side, "attacker" and "attacker nodes" refer to those with malicious intentions. To achieve their goals, attackers modify the rules within their local copy of the software, attempting to undermine the integrity of the network. So, while cryptography and incentives help make attacks more complicated, the expectation that the majority of nodes will act honestly keeps Bitcoin secure. As the Bitcoin white paper states, "the proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-computational power-one-vote". [15] The network's integrity is upheld through the honest participation of nodes, which collectively control more computational power than any cooperating attackers. But for this, they must continuously occupy a significant amount of computational hardware and electricity, constantly increasing their usage and raising the capital spent on keeping the system secure. This results in inefficiency where massive resources are allocated not toward generating value (in Bitcoin, 0% of incentives are directed towards participants performing productive tasks, as the entirety of rewards is consumed by the validation of non-productive computations) but rather maintaining a dominance of computational power, making security dependent on an ever-growing consumption of energy and hardware, with diminishing returns. In reality, the only thing Proof-of-Work is trying to achieve is to ensure that the honest majority has more votes in determining what is true.

Proof-of-Stake

The blockchain community has explored Proof-of-Stake as an alternative to Proof-of-Work, primarily to mitigate the cost of computing and the environmental impact associated with Proof-of-Work consensus mechanism. In Proof-of-Stake systems, participants lock up a certain amount of cryptocurrency in a frozen account, with voting power and validation responsibilities distributed based on the amount staked.

While Proof-of-Stake aims to maintain network security with lower energy use, it introduces significant capital costs associated with staking. In Proof-of-Stake systems, incentives are primarily directed toward capital holders rather than contributors to computational tasks, resulting in high costs for the network. Modern cryptocurrency systems often compensate this capital at rates significantly above market levels. For example, Bittensor currently offers an IRR of 18% (excluding the increase in coin market prices). In Bittensor, 60% of additional incentives go to those who do not contribute to computational work, clearly illustrating the inefficiency of this model. If Ethereum had been a Proof-of-Stake network from the start, nearly all of its $300 billion market capitalization (aside from a percentage allocated to ICO) — would

have been paid as subsidies to capital rather than being invested in hardware subsidies. This capital could have been used to significantly advance AI by subsidizing substantial computational infrastructure.

Additionally, Proof-of-Stake systems only ensure an honest majority for maintaining the ledger, not for the execution of utility tasks. For example, while Proof-of-Stake-based networks like Ethereum provide consensus on the state of the ledger, they do not verify the correctness of off-chain AI computations. This lack of execution-level trust often leads to redundant re-computation for result verification. Unlike deterministic smart contract logic, AI tasks are computationally intensive and often probabilistic, making repeated execution wasteful and unsustainable.
AI applications, such as autonomous vehicles and live analytics, require fast responses. Repeating tasks increases latency, making the network unsuitable for real-time demands. Also, repetition amplifies energy and computing resource consumption, leading to higher costs and reduced scalability in decentralized networks with limited resources.
In decentralized AI networks, it is crucial to extend this honest majority to task execution to avoid unnecessary task repetitions and resource waste.

Transition from Proof-of-Work to Proof-of-Stake

The shift from Proof-of-Work to Proof-of-Stake allowed the development of new blockchain networks designed for real-world applications, like Bittensor. Proof-of-Stake incorporates a key element where only one chain and a block are considered valid when the majority of participants sign off on it, ensuring consensus through validator agreement. By removing the need for energy-intensive mining, Proof-of-Stake enabled systems that could allocate resources toward tasks beyond transaction validation. Bittensor uses Proof-of-Stake to power a decentralized marketplace where participants contribute to AI tasks, such as training models, in exchange for rewards.

However, while this shift enabled the network to focus on solving meaningful problems, it also sacrificed the fairness and decentralization of Bitcoin's Proof-of-Work system. In Proof-of-Work, rewards are tied directly to computational effort, ensuring that anyone with the necessary hardware can participate and earn rewards. By contrast, Proof-of-Stake ties influence token holdings, meaning those with more capital control more of the network. This creates a system where wealth concentration can distort reward distribution, limiting true decentralization and fairness.

Proof of Useful Work

Recognizing these challenges, some projects are exploring Proof of Useful Work to combine the benefits of Proof-of-Work and Proof-of-Stake while improving resource efficiency. Proof of Useful Work aims to maintain the decentralization and fairness seen in Proof-of-Work but directs the computational resources toward solving practical problems. Rather than simply validating transactions, Proof of Useful Work systems could allocate computing power to tasks like AI model training, protein folding, or other scientific computations, turning the work into something useful beyond network maintenance.

Early attempts at implementing Proof of Useful Work show promise in integrating valuable tasks with blockchain consensus. However, technical hurdles remain, such as ensuring the usefulness and measurability of these tasks while keeping the system decentralized.

4. Proof-of-Work reimagined: a new approach to efficient utilization of computational power

While Proof-of-Stake offers an alternative to Proof-of-Work by aiming to reduce computational cost and energy consumption, it introduces challenges related to resource allocation and network efficiency. This has led to reconsidering what defines an effective and equitable consensus mechanism in decentralized networks. In crypto projects, truth is established through protocols designed to ensure that the majority genuinely represents the majority and is not merely an illusion created by a deceptive minority. This raises a fundamental question: How could the majority be determined effectively?

The proposed novel consensus algorithm combines both key elements of Proof-of-Work and Proof-of-Stake to create a balanced and efficient network security and governance mechanism. It incorporates the key element of Proof-of-Stake, where only one chain and a block are considered valid when the majority of participants sign it off. However, unlike the traditional Proof-of-Stake mechanism, where voting weight is based on the amount of cryptocurrency staked, in the proposed system, voting weight is determined by the amount of computational evidence generated during a competitive Race, similar to the Proof-of-Work process. The Race is a structured, time-bound competition where participants utilize their computational resources to solve complex tasks within a defined timeframe. The outcomes of these tasks serve as evidence, determining the voting weight each participant holds in the network.

This new approach returns to the roots of Proof-of-Work, focusing on computational power but optimizing its use. It achieves efficiency by recognizing that the primary purpose of computational tasks (ensuring network security and fairness) can be accomplished within a limited timeframe. During this limited timeframe, participants fully utilize their computational power to generate evidence, which is then used to determine voting weight. This method frees up computational resources beyond the Race timeframe, so that they can be directed toward meaningful tasks, enhancing the efficiency and sustainability of decentralized AI networks. *For a technical specification of the Proof-of-Work mechanism designed for this framework, refer to Appendix A.*

During the Race, all participants engage in a computationally demanding task designed to be resource-intensive but easy to verify (described in more detail in Appendix A). Similar to Bitcoin's Proof-of-Work nature, where finding the correct nonce to produce a hash with one leading zero requires about 16 attempts, achieving a hash with ten leading zeroes could require trillions of attempts. Once the correct nonce is found, any node can easily verify its correctness.

Following the Race, nodes (holding voting weights from the previous Race or from the genesis block in the case of the first Race) record new weights based on the result of the recent Race. These weights are then utilized to evaluate whether consensus is reached for a specific block candidate. Instead of basing voting weight on the amount of staked cryptocurrency - as in Proof-of-Stake the system allocates voting weight based on the quantity of valid computational work each participant performed during the Race, employing a "one-computational-power-one-vote" principle outlined in the Bitcoin whitepaper. The proportion of voting weight in the proposed system is equivalent to what is observed in standard Proof-of-Work systems, where voting weight is directly tied to the computational effort exerted. Unlike traditional Proof-of-Work systems, where computing wastes 100% of the time finding hash functions, the proposed method utilizes computational power for meaningful tasks, such as AI model training and inference. The allocated voting weights remain effective until the end of the cycle, after which a new Race determines the next set of weights.

Unlike other decentralized computational networks, in the proposed protocol, the resulting voting weights are not only used to maintain ledger records but also to distribute meaningful tasks between

computational nodes and validate the results returned by these nodes. This approach significantly reduces the need for redundant computations. This is possible because, unlike in PoS, voting weights correlate with the actual computational capacity of the nodes, and unlike in PoW, computational capacity is not occupied by meaningless tasks. The details of this process are described…

This method of allocating voting weights directly influences how rewards are distributed within the network. Rewards are strictly allocated to participants who contribute to meaningful tasks, such as AI model training and inference. This ensures that participants are incentivized not just to secure the network but to contribute to its growth and functionality actively.

The Race

Race is a critical component of the novel consensus mechanism. It begins simultaneously for all participants, ensuring fairness by eliminating any advantage based on timing. This synchronous start is akin to a "starting gun" in a race, where no participant is allowed to start earlier than the others. To ensure this, a random number that cannot be predicted or influenced in advance is generated by all participants with voting power. This randomness is essential for preserving the integrity of the process, preventing any minority group of participants from pre-computing or gaining an unfair advantage. *Appendix B outlines the process for generating this random number, which is vital for maintaining fairness and security in the decentralized AI network.*

The duration of the Race is deliberately confined to a short, defined timeframe, typically around 10 minutes, ensuring that the computational effort is both concentrated and efficient. The Race occurs at regular, precisely specified intervals on a cycle basis, synchronized with the creation of blockchain blocks (e.g., every k*n-block in the blockchain). This regularity integrates the Race seamlessly into the network's operational rhythm.

The Task

The computational power loading function is specifically designed to favor participants with hardware optimized for tasks similar to those required for training LLMs. To achieve this, the computational task executed during The Race closely mirrors the structure of transformer calculations, ensuring that the participants who succeed are those equipped with the appropriate hardware for such advanced computations. *For a detailed explanation of the Proof-of-Work mechanism designed for this framework, which aligns computational power with AI workloads, please refer to Appendix A.*

The Race must not interfere with normal user requests. When user requests are present, they must be processed without delay. In the absence of user requests, to ensure full computational power utilization, the system executes an alternative function, or "dummy tasks," whose outcomes cannot be predetermined. This is analogous to Bitcoin's hash function calculations, which serve no purpose beyond securing the network.

5. Real-world utility

Using the proposed protocol, a decentralized network can emerge, where participants contribute computational resources to the network ("Hardware Providers"). Their hardware will run software based on the described protocol ("nodes"). According to the protocol as described above, each participant will get assigned a voting weight corresponding to its actual computational power. Their nodes will continuously reach consensus and maintain a ledger of records similar to how it's done in networks like Bitcoin or Ethereum. At the same time, their computational power remains available for real-world utility tasks such as AI inference or training.

Hardware providers will be compensated (either by other participants or through the issuance of new cryptocurrency) for executing tasks, covering their costs, and generating profits to sustain their operations. The distribution of computational tasks is proportional to the processing power of each participating node, and the rewards are allocated accordingly. This ensures that hardware providers receive a fair level of utilization and, consequently, a fair share of the rewards. *For an example of the reward system for AI inference tasks, see the Chapter "Incentives".*

However, in decentralized networks, we cannot assume all nodes are honest nor that they won't attempt fraudulent behavior. The following types of fraud are possible when nodes perform tasks:

- A node may fail to complete the assigned task, hoping to receive a share of the reward without contributing. The node could either fail to return a result or return a result inconsistent with what would have been generated by genuine work.
- A node might return a result using a simpler model (e.g., with fewer parameters), hoping to receive full compensation while reducing its actual costs.
- A node may attempt to insert misinformation, malicious manipulation, or advertising into the returned result.

To mitigate these risks, the system incorporates a set of countermeasures designed to ensure efficiency without causing excessive computational overhead.

6. Task verification

Majority Verification

In decentralized networks, the results produced by individual nodes cannot be trusted without verification. When other participants perform a verification, the assumption should be made that there is always a risk of collusion between the nodes that produce the result and the ones verifying it. To address this, the system relies on a principle of majority verification based on the voting weight of computational nodes. Task verification is sufficient when nodes representing more than 50% of the total voting weight confirm the result, as we rely on the assumption that the majority of participants, based on their voting weight, are honest. This approach provides a clear guideline on how many verifications are required and when the results can be 100% trusted.

This method reduces the number of redundant verifications compared to systems where every node must verify every task. However, even with this approach, multiple verifications are still required to maintain security and integrity, introducing a significant verification overhead. Randomized Task Verification is introduced to further optimize resource use, which complements majority verification by drastically lowering the number of necessary checks while maintaining trust in the network.

Reward accumulation and distribution

Before delving into randomized task verification, it is important to understand how rewards are accumulated and distributed in the system. Since not every task can be verified instantly, rewards for completed tasks accumulate during the cycle between races, awaiting full verification. The release of these rewards depends on whether the node has fulfilled three main criteria:

1. **Accuracy:** The node must not have been found guilty of producing false or malicious results.
2. **Work Proportion:** The node must complete its fair share of the total workload.
3. **Validation Proportion:** The node must fulfill its role in the validation process by verifying a proportional amount of work done by others.

These criteria ensure that nodes are incentivized to complete both computational and validation tasks honestly and efficiently. If a node fails to meet any of these requirements, its rewards may be reduced or forfeited entirely. This reward accumulation system protects the network from dishonest behavior by holding rewards in reserve until verification is complete, ensuring that rewards are only distributed for tasks that have been successfully and honestly validated.

With reward accumulation in place, the system is equipped to implement randomized task verification, a probabilistic approach designed to reduce the verification burden while maintaining security and integrity.

Randomized task verification

To further minimize verification costs without compromising trust, the system employs Randomized Task Verification. Rather than verifying every task, the network randomly selects a subset of tasks for verification. This process ensures that nodes cannot predict which tasks will be checked, thereby discouraging malicious behavior.

Work produced by one node is subject to verification by other nodes, but this verification follows a random pattern akin to spot-checks. Just as parking violations are enforced through random checks, where compliance is encouraged by the risk of being penalized, nodes in the network are subject to the same uncertainty regarding task verification. Malicious nodes risk losing all accumulated rewards if caught, making it more attractive to comply with the network's requirements.

Each participant's voting weight determines the probability that they will be assigned a verification task. For example, a node with 50% of the voting weight may be responsible for verifying 1 out of every 20 task results produced by a node, while a node with 10% of the voting weight may verify one out of every 100 task results. This system ensures that, collectively, participants verify approximately one out of every 10 tasks, with the majority of nodes verifying one out of every 20 tasks within their cluster. Even if some verifiers are malicious, the honesty of the majority ensures that every task is eventually verified by a trustworthy participant, preserving the network's integrity. If a node is caught producing false results, it forfeits all rewards earned during that cycle.

A probabilistic model ensures that any malicious node will eventually be caught over a series of tasks. Since nodes perform numerous tasks within each cycle, and rewards are distributed only at the end of the cycle, the likelihood of a malicious node going undetected is minimized. Even if a few false results escape detection, the cumulative probability of catching malicious behavior over time makes cheating unprofitable.

Allocating votes based on computational power rather than capital ensures that nodes contributing computational resources are effectively able to validate the work of others. This method reduces the resource waste typically associated with redundant checks in current decentralized utility systems, lowering the required repetition rate to as little as 1-10%.

The validation process is further strengthened by a pseudo-random algorithm that governs how nodes select which tasks to validate. This system ensures that the decision to validate a particular task is not purely random but is instead influenced by deterministic factors, making it possible to audit the selection process. For instance, each transaction is associated with a unique ID. A node applies its private key to sign this ID, generating a signature that serves as a seed for determining whether the node should validate the specific task. Importantly, this signature and the validation result can be shared with the network. Other participants can independently verify that the signature is authentic and that it correctly served as the seed for the random function, which determined that the node was supposed to validate that specific

task. This level of transparency ensures that all nodes adhere to their responsibilities, allowing the entire network to maintain trust in the validation process.

Combining majority verification and randomized task verification, the system balances trust and efficiency, ensuring that tasks are verified rigorously while minimizing resource waste. Rewards are only distributed to nodes that have passed all verifications, further reinforcing the importance of honest participation in the network.

Verification challenges

The process of validating whether a node is malicious when it comes to AI inference or training presents significant technological challenges, particularly due to the non-deterministic nature of some hardware operations. Hardware tends to produce varying results even when using the same input and randomization seed. As a result, determining whether a node is acting maliciously cannot be treated as a purely deterministic task; it must be approached as a statistical problem. This recognition means that conclusions about whether a node is engaged in a malicious activity are based on probabilities rather than certainties, with considerations for both false positives and false negatives.

When a result is flagged as potentially incorrect, it cannot be immediately assumed to be malicious. Instead, other nodes, which together represent the majority of voting weight, must re-verify the result. This re-verification process takes into account the potential variability in outcomes, recognizing that even honest nodes may produce slightly different results due to hardware differences.

This statistical approach to verification is particularly important in reducing the likelihood of false positives. The final decision regarding a node's honesty might involve accepting a small number of false positives, allowing nodes to make a limited number of errors before facing penalties. In this way, punishment is reserved for nodes that consistently exceed an acceptable error threshold rather than being triggered by a single mistake. *A potential implementation for the verification process is described in more detail in Appendix A.*

Reputation

Reputation can accumulate over time as nodes continue to act honestly. New participants start with a reputation score of zero. As a participant successfully contributes to the network without being caught producing false results, their reputation increases with each subsequent cycle. The longer a participant remains honest, the higher their reputation grows.

A low reputation score increases the likelihood that a participant's work will be scrutinized — up to the point where every single task they produce may be verified. Conversely, participants with a higher reputation face less frequent checks. The reward a participant receives thus depends on how much of their work is verified by others. For example, if every 10th task is verified, a participant with a high reputation receives 100% of the rewards for nine tasks and only 50% for the one that is verified. If every task is verified, the reward is halved. If a participant is caught producing malicious work, they lose all rewards for that period, and their reputation is reset to zero, placing them under the highest level of scrutiny again.

Dummy tasks

During periods of low user activity, the system may introduce mechanisms to allow nodes to build a reputation by performing tasks that are not directly generated by real users. These "dummy tasks" are designed in two ways:

1. **Distinguishable Dummy Tasks:** Nodes differentiate real user requests and dummy tasks, ensuring that they allocate resources appropriately.
2. **Indistinguishable Dummy Tasks:** Alternatively, dummy tasks can be designed to be indistinguishable from real user requests, making it impossible for nodes to prioritize or neglect tasks based on their origin. These prompts must be generated by an algorithm that closely mimics real user behavior, preventing nodes from discerning whether the task is genuine or synthetic. Indistinguishable Dummy Tasks, designed to ensure that nodes cannot prioritize or neglect tasks based on their origin, present unique challenges that must be addressed to maintain the integrity and efficiency of the network. These challenges are centered around ensuring task authenticity, anonymity, and fair distribution.

Below are the three primary issues that arise with the implementation of Indistinguishable dummy tasks:

1. **Indistinguishability of prompts nature:** The first challenge lies in ensuring that dummy tasks are indistinguishable from real user requests. This requires the algorithm that generates these prompts to be sophisticated enough that nodes cannot differentiate between real and synthetic tasks. The generation process must be so robust that even with extensive analysis, nodes cannot confidently assert whether a user request is genuine or a dummy task. If nodes were able to distinguish between these tasks, they might allocate resources differently, leading to an imbalance in the system and potentially undermining the fairness of the network.
2. **Unlinked transaction balances:** The second challenge involves the financial transactions associated with these tasks. All transactions, whether linked to real user requests or dummy tasks, must carry a balance that is not affiliated with the node executing the task. This is to ensure that the incentive mechanism remains unbiased and anonymous. The protocol needs to account for the back compensation of funds to the entity generating dummy tasks. However, this compensation must be processed through random accounts rather than directly from the main account, preventing any association with specific nodes. This level of financial decoupling is critical to maintaining the anonymity and fairness of task allocation.
3. **Anonymizing task origin:** The third challenge is related to the anonymity of the task's origin. To preserve user privacy and prevent nodes from identifying the source of the request, the network is structured so that user requests are first routed through a random node. This node does not perform the task itself but instead determines which participant should complete it according to the network protocol. This mechanism ensures that the majority of user requests are anonymized, with the originating user's identity obscured. By randomizing the node that initially receives the task and ensuring that this node acts as an intermediary rather than a direct executor, the system effectively hides the user's IP address and further secures their anonymity.

User-driven oversight

The system also includes a mechanism for end-user-driven oversight. If a user suspects that a result is incorrect or suspicious, they can submit a paid report by clicking a "dislike" or "report" button. This action triggers a verification process by other nodes. If the user's report is validated, the malicious node forfeits its rewards, which are then shared with the user and the verifying nodes. However, if the report is found to be incorrect, the nodes involved in the verification process are rewarded instead. This mechanism ensures that the system heals itself by continuously reinforcing the network's integrity and trustworthiness through participant and user oversight.

Workload proof

Finally, the system assumes that both workload and verification duties are evenly distributed among participants. It is penalized if a node fails to complete its proportional share of tasks or verification duties. The specific penalties are determined by a formula that accounts for the node's overall performance

relative to the network's expectations. This ensures that every participant contributes fairly to the network's operation, maintaining the balance and integrity of the decentralized system.

Bounty reward

The rewards for identifying malicious nodes must be carefully calibrated to ensure they do not exceed potential earnings from honest participation. This precaution prevents unintended incentives that might arise from discovering and penalizing malicious nodes.

Moreover, if a node is found to be engaging in malicious behavior when it processes a request paid by a user, the payment should be reimbursed, so it can't be used as part of the bounty reward. This consideration underscores the importance of maintaining user trust and ensuring the system remains equitable for all participants.

7. Inference

A network built on the principles described here can offer external developers LLM inference services using widely adopted open-source LLM models, such as Llama 3.2 70B or Mistral Large.

Developers submit their users' requests to the network, which are then processed by the nodes of hardware providers using the specified model to generate responses *(see Appendix D for details on Nvidia GPUs and their suitability for inference tasks)*. For developers' convenience, these requests can be made through an API similar to those provided by centralized services like OpenAI Platform.

Key participants in a decentralized AI ecosystem for LLM inference

A decentralized AI infrastructure relies on the collaboration of several key participants:

- Hardware providers (participants or nodes). These participants contribute computational resources to the network and are rewarded based on the amount and quality of resources they provide.
- Developers. Developers build and deploy AI applications within the decentralized network, leveraging the distributed computational power to run their models. This approach provides developers greater flexibility and reduced costs, as cloud providers' limitations and pricing models or closed-source LLMs do not constrain them.
- Users. End-users interact with AI applications running on the decentralized network. Their requests (inferences) generate the demand for computational resources, driving the growth and expansion of the ecosystem.

8. Training and fine-tuning

Motivation and the Current State of Training

Distributed machine learning faces multiple critical challenges that impede truly decentralized training at scale.

1. When participants join a distributed training network, they are fundamentally untrusted entities, with some potentially submitting fraudulent computations or malicious data. Any viable system must be able to identify these bad actors and implement appropriate penalties.
2. Unlike controlled data centers with high-bandwidth internal networks, decentralized training must operate over standard internet connections with their inherent bandwidth limitations and latency issues.

3. Traditional approaches to training large models require each participant to make gradient steps for the entire model locally—an unrealistic expectation given the demand for massive computational resources from every participant.
4. To date, most distributed training approaches have relied on centralized coordinator nodes, introducing single points of failure and trust concerns.

Our approach aims to address these challenges through several solutions. We leverage DiLoCo's communication-efficient training approach [18] to significantly reduce bandwidth requirements between participants. We replace centralized coordination with on-chain management, eliminating single points of failure and enhancing system resilience. To address the issue of untrusted participants, we implement a proof-of-learning [19] inspired validation system that enables verification of training contributions. For scaling model size without requiring participants to store the entire model, we explore sharding approaches that allow participants to host only portions of the complete model.

Each of these solutions is described in further detail in Appendix M.

To ensure long-term sustainability of these efforts, the network dedicates 20% of all inference revenue to support future model training. This guarantees that foundational AI development continues even beyond the early subsidized stages, aligning economic incentives with the evolution of the ecosystem. *More on hardware providers' compensation model for training new Large Language Models is in Appendix K.*

Fine-tuning

Fine-tuning provides developers with the ability to fine-tune open-source models to meet specific tasks or applications. This process builds on the existing foundational models, allowing developers to submit targeted data and request fine-tuning, similar to established practices in platforms like OpenAI. Participating nodes in the network use the provided data to conduct additional training, forming supplementary layers that tailor models even further to meet developers' needs, achieving improved results without the need for extensive prompts. These enhancements contribute to more efficient token usage and lower latency during inference. The decentralized nature of the network brings collaborative expertise into the fine-tuning process, fostering varied approaches that enhance model performance. This fine-tuning framework ensures that developers can leverage enhanced, specialized versions of models for their unique purposes, maintaining flexibility while benefiting from the strengths of a decentralized infrastructure.

9. Incentives

Dual-reward system

A dual-reward system analogous to the mechanisms observed in Bitcoin is proposed to incentivize participation within the network. Participants who provide computational power are compensated through two primary avenues: transaction fees and a network-wide inflationary reward system.

Transaction fees are embedded within each transaction, similar to Bitcoin, where the end-user or developer attaches a fee to execute specific tasks. This ensures that nodes contributing computational resources are rewarded directly for their efforts.

In addition to transaction fees, participants receive rewards through an inflationary mechanism. Each block generated within the network issues a reward distributed proportionally among participants based on their computational contribution. This reward decreases over time, following a phased approach:

1. Initial subsidy phase: A substantial subsidy offsets the capital expenditure (CAPEX) associated with providing hardware in the network's early stages. This incentivizes early participation and ensures network growth. This subsidy will taper off as the network matures, transitioning the primary reward source to transaction fees, mirroring Bitcoin's diminishing block rewards.
2. Market efficiency: As subsidies decrease, market efficiency is expected to rise, driven by advancements in hardware performance. Just as the Bitcoin network saw the development of highly efficient ASICs, the network anticipates similar innovations, reducing the cost of computational power provision and ensuring sustainability as rewards shift entirely to transaction fees.

This dual-reward system also addresses the challenge of hardware utilization. In the network's launch phase, when hardware may be underutilized, initial subsidies distribute costs across fewer tasks, making CAPEX more manageable. As the network expands and hardware utilization increases, the need for subsidies diminishes, and transaction fees become the sole reward source. This progression allows the network to scale effectively while maintaining a competitive edge over centralized cloud providers, ultimately offering services at a fraction of the cost.

Transaction cost management

Transaction costs within the network are designed to be flexible and predictable, akin to Ethereum’s GAS model or the OpenAI API's pricing structure. Users specify a maximum cost they are willing to pay, ensuring that tasks are performed within this predefined limit.

If computational resources are exhausted before task completion, the transaction is canceled, and the spent tokens are not refunded, similar to Ethereum’s handling of exceeded GAS limits. The rewards generated from these transactions are distributed between the node executing the task and the one verifying it, ensuring fair compensation across participants.

Long-term sustainability

The long-term strategy focuses on attracting participants willing to dedicate their hardware to the network, positioning the project as a cost-competitive alternative to traditional cloud providers. Leveraging decentralization, the network is designed to offer significantly lower prices for computational tasks, making it an attractive option for developers and businesses seeking cost-effective solutions.

In conclusion, the tokenomics strategy is crafted to ensure the network's sustainability, support early adopters, and maintain competitive pricing. Balancing initial subsidies with a gradual transition to transaction fees establishes a robust ecosystem that rewards participants, drives hardware innovation, and positions the network as a leader in decentralized AI computation.

10. Conclusion

This paper has introduced a decentralized AI infrastructure utilizing the “Proof-of-Work 2.0” consensus mechanism, and it improves both centralized and current decentralized systems by addressing their core inefficiencies.

Current centralized AI infrastructures, dominated by a few major cloud providers, suffer from high operational costs, monopolistic pricing, and risks of censorship. These limitations restrict access to advanced AI technologies, especially for smaller developers, while concentrating control in the hands of a few entities. In contrast, the proposed novel decentralized model democratizes access to AI resources,

fostering innovation through competitive pricing. Unlike existing decentralized systems that rely heavily on inefficient consensus mechanisms such as traditional Proof-of-Work or Proof-of-Stake, where the majority of computational power is misdirected towards securing the network rather than performing valuable tasks, the proposed system ensures almost 100% of computational resources are used for meaningful AI tasks like training and inference. This approach eliminates resource waste in systems such as Bitcoin, where all incentives are consumed by network security, or Bittensor, where only 40% of rewards go towards AI computation. The proposed system maximizes hardware utilization by aligning computational efforts directly with productive outputs, significantly reducing the cost of operating AI models.

The novel Proof-of-Work 2.0 mechanism balances the strengths of both Proof-of-Work and Proof-of-Stake systems while eliminating their drawbacks. It adopts the fairness and security of Proof-of-Work, ensuring computational power is utilized effectively while avoiding the capital inefficiencies of Proof-of-Stake, where incentives disproportionately benefit capital holders rather than computational contributors. This hybrid model rewards participants solely for their contributions to AI workloads, ensuring that every unit of energy spent yields productive results.

The decentralized structure eliminates the monopolistic control associated with centralized cloud providers. Distributing computational tasks across a global network of nodes offers greater resilience, transparency, and censorship resistance. Developers' ability to deploy AI applications on this decentralized infrastructure without the constraints of centralized pricing or policy control ensures more flexibility and autonomy, particularly for smaller developers traditionally priced out of the market by centralized cloud giants.

The proposed decentralized AI system offers a better alternative to centralized and traditional decentralized systems. It reduces costs, increases resource efficiency, ensures meaningful use of computational power, and democratizes access to advanced AI capabilities.

References

[1] Slingerland, Cody. “Azure Vs. AWS Pricing: The Ultimate 2024 Guide.” CloudZero, 8 October 2024, https://www.cloudzero.com/blog/aws-vs-azure-pricing/. Accessed 22 October 2024.
[2] Gil, Laurent. “Cloud Pricing Comparison: AWS vs. Azure vs. Google Cloud Platform in 2024.” CAST AI, 13 December 2023, https://cast.ai/blog/cloud-pricing-comparison-aws-vs-azure-vs-google-cloud-platform/. Accessed 22 October 2024.
[3] Muñoz, Andres. “AWS vs Azure Pricing: A Complete Comparison.” Nacho Nacho, 25 July 2024, https://blog.nachonacho.com/tech/aws-vs-azure-pricing/. Accessed 22 October 2024.
[4] Compare LLM API Pricing Instantly - Get the Best Deals at LLM Price Check, https://llmpricecheck.com/. Accessed 22 October 2024.
[5] Kahn, Jeremy, et al. “Meta's new 405 billion parameter Llama 3.1 model could be a game changer.” Fortune, 23 July 2024, https://fortune.com/2024/07/23/meta-new-llama-model-3-1/. Accessed 22 October 2024.
[6] Jones, Mike. “Microsoft Azure Negotiation Strategy – Large Deals.” US Cloud, 31 July 2023, https://www.uscloud.com/blog/microsoft-azure-negotiation-strategy-large-deals/. Accessed 22 October 2024.
[7] “Do the costs of the cloud outweigh the benefits?” The Economist, 3 July 2021, https://www.economist.com/business/2021/07/03/do-the-costs-of-the-cloud-outweigh-the-benefits. Accessed 22 October 2024.
[8] Erten, Tugce. “The Cost of Cloud, a Trillion Dollar Paradox.” Andreessen Horowitz, 27 May 2021, https://a16z.com/the-cost-of-cloud-a-trillion-dollar-paradox/. Accessed 22 October 2024.
[9] Lutz, Sander, and Mat Di Salvo. “Is Solana Decentralized? Cloud Provider Hetzner Ban Raises Questions.” Decrypt, 3 November 2022, https://decrypt.co/113429/is-solana-decentralized-cloud-provider-hetzner-ban-raises-questions. Accessed 22 October 2024.
[10] Schneier, Bruce. “Censorship in the Age of Large Cloud Providers.” Lawfare, 7 June 2018, https://www.lawfaremedia.org/article/censorship-age-large-cloud-providers. Accessed 22 October 2024.
[11] “The Juno White Paper”, https://juno.build/docs/white-paper/problems-statement. Accessed 22 October 2024.
[12] “The Battle for Online Freedom: Centralization vs. Decentralization (Part 3).” Media Foundation, 13 March 2023, https://mediafoundation.medium.com/the-battle-for-online-freedom-centralization-vs-decentralization-part-3-5512510a1aeb. Accessed 22 October 2024.
[13] “A Proof of Useful Work for Artificial Intelligence on the Blockchain.” arXiv, 25 January 2020, https://arxiv.org/pdf/2001.09244. Accessed 22 October 2024.
[14] Peters, Keaton, Katie Surma, and Marcos Colón. 2024. “US Government Launches New Attempt to Gather Data on Electricity Usage of Bitcoin Mining.” Inside Climate News. https://insideclimatenews.org/news/11072024/us-energy-information-agency-bitcoin-mining-electricity-usage-data/
[15] “A Peer-to-Peer Electronic Cash System.” n.d. Bitcoin.org. Accessed August 9, 2024. https://bitcoin.org/bitcoin.pdf.
[16] Bitcoin Block Reward Halving Countdown, Accessed 5 December 2024, https://www.bitcoinblockhalf.com/.
[17] “How The Merge impacted ETH supply”, Accessed 5 December 2024, https://ethereum.org/en/roadmap/merge/issuance/
[18] Douillard, Arthur, et al. "Diloco: Distributed low-communication training of language models." arXiv preprint arXiv:2311.08105 (2023).
[19] Jia, Hengrui, et al. "Proof-of-learning: Definitions and practice." 2021 IEEE Symposium on Security and Privacy (SP). IEEE, 2021.

[20] McMahan, Brendan, et al. "Communication-efficient learning of deep networks from decentralized data." Artificial intelligence and statistics. PMLR, 2017.
[21] Reddi, Sashank, et al. "Adaptive federated optimization." International Conference on Learning Representations (2020).
[22] Lepikhin, Dmitry, et al. "Gshard: Scaling giant models with conditional computation and automatic sharding." arXiv preprint arXiv:2006.16668 (2020).
[23] Ryabinin, Max, and Anton Gusev. "Towards crowdsourced training of large neural networks using decentralized mixture-of-experts." Advances in Neural Information Processing Systems 33 (2020): 3659-3672.
[24] Douillard, Arthur, et al. "Dipaco: Distributed path composition." arXiv preprint arXiv:2403.10616 (2024).
[25] Charles, Zachary, et al. "Communication-Efficient Language Model Training Scales Reliably and Robustly: Scaling Laws for DiLoCo." arXiv preprint arXiv:2503.09799 (2025).

Appendix A

Transformer-Based Proof-of-Work: Aligning computational power with AI workloads

General description

The proposed Proof-of-Work mechanism is designed to leverage the computational characteristics of neural network architectures, effectively aligning the voting power with the actual workload of LLMs in a decentralized AI network. Participants are incentivized to optimize their hardware and software for LLM computations.

The Proof-of-Work algorithm is structured to resemble the computational capacity for LLM inference and training. Every node executes a parametrized function comprising computational blocks typical to LLM architecture illustrated in Figure 1. The computational capacity of a node is demonstrated through the number of vectors found during the Race, that satisfy a specific condition. A proof of computational capacity is sent to the network and validated by other nodes. Voting power in the network is directly proportional to the computational capacity demonstrated during the Race, ensuring fair representation based on actual contributed resources.

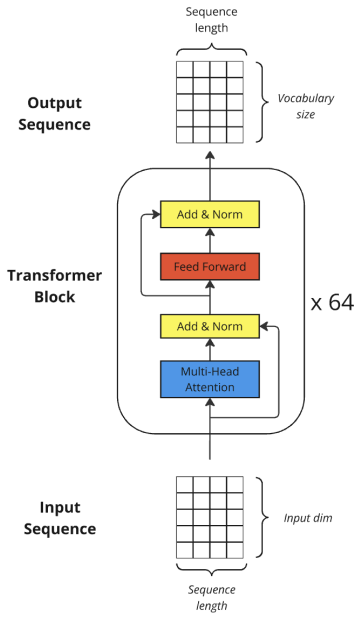


Figure 1. Transformer architecture illustration

Participants are incentivized to optimize their hardware and software for LLM computations, benefiting both the consensus mechanism and the network's primary function of AI processing. The randomness of the generated model for each Race prevents pre-computation advantages.

Race procedure

1. **Initialization**

- **Race seed generation.** At the start of each Race cycle, a **Race Seed** is generated with a random number generator based on the latest blockchain state. This seed is used to initialize the shared parameters of the parametrized function. The Race Seed is specific to each race and the same across all the devices participating in that Race.
- **Model initialization.** The parametrized function is a compact Transformer-based model (hereinafter referred to as Transformer) imitating LLMs on a smaller scale. This Transformer consists of 64 layers, each containing a multi-head attention mechanism with 128 heads and a feed-forward block. The model has an embedding dimension of 512 and uses a vocabulary size of 8192. The feed-forward block in each layer expands to a hidden dimension of 8192 before projecting back to 512. The model is designed to process sequences of length 4. Total number of parameters is about 2.3 billion.
- **Node Seed creation.** After the model is initialized, every node generates its unique Node Seed that is based on its public key.

2. **Iterations**
   - **Nonce iteration.** During the Race, every participant iterates over nonce values (Input Seeds), which are combined with Node Seed and a current Race Seed produce input sequences to the Transformer. These sequences are passed to the Transformer to compute the output sequences, and only the last vectors of output sequences are used to construct a proof.
   - **Appropriate vectors.**
     i. Participants must find output vectors close enough to a **Target Vector** in terms of Euclidean distance. Vectors satisfying the described condition are called **Appropriate Vectors**.
     ii. The **Target Vector** is randomly initialized at the beginning of each Race with a Race Seed and is common for all the nodes participating in the Race.
   - **Distance threshold**. The distance threshold is constant and is chosen such that the chance of finding an Appropriate Vector from one nonce is about 1 in 900 and can be used to control difficulty.
3. **Fraud prevention**
   - **Random permutation.** To prevent fraud, the output vector's coordinates are randomly permuted right before the calculation of the distance to the Target Vector. This procedure ensures that a node can't abuse the continuity of the neural network, in particular, if a node finds an Appropriate Vector, it could just iterate over nonce values to find an input vector close to the one that produced the Appropriate vector.
   - **Defining factors**. The nonce value, Node Seed, and Race Seed define this random permutation.
4. The more Appropriate Vectors are found, the more times the Transformer was inferred during a fixed time duration Race, which indicates higher computational capacity. This condition implements a similar principle to the Bitcoin Proof-of-Work. Finding an Appropriate Vector is analogous to finding the hash with a particular number of leading zeros.

Validation of proof

The proof of computational capacity is a set of nonce values. All of these nonce values should produce such input sequences that, if entered into the Transformer, they produce Appropriate Vectors.

The proof validation process consists of 3 steps:

1. Reconstruction of the Transformer and the Target Vector of the race with the Race Seed, using the blockchain state at the time of the Race.
2. Generation of the node's input sequences using the provided nonce values and the Node Seed.
3. Performing Transformer forward passes, and confirming that all the provided nonce values result in Appropriate Vectors.

Voting weight is allocated proportionally to the number of valid proofs presented by the participant.

By implementing this LLM PoW mechanism, the decentralized AI network achieves a harmonious balance between security, fairness, and practical utility, positioning itself at the forefront of efficient and purpose-driven blockchain consensus for AI applications.

Code examples

The examples below illustrate how these procedures might look in Python-like pseudocode, with some functions serving as conceptual placeholders rather than executable code. Race duration and distance thresholds are fixed constants.

```python
def generate_proofs(node_public_key, latest_blockchain_state):
    """
    Generates valid nonces.
    For each nonce, the function generates input sequence and checks if
    this input sequence lead to an appropriate vector for the current race.
    Every nonce that generate an appropriate vector is sent as a proof.
    Args:
        node_public_key (str): Node's public key.
        latest_blockchain_state (str): Current blockchain state.

    Returns:
        list : Valid nonces.
    """
    timer.start()
    race_seed = generate_race_seed(latest_blockchain_state)
    transformer_model = initialize_transformer(race_seed)
    node_seed = generate_node_seed(node_public_key)
    target_vector = generate_target_vector(race_seed)
    valid_nonces = []
    for nonce in nonce_generator:
        input_seed = combine_seeds(nonce, node_seed, race_seed)
        input_sequence = generate_input_sequence(input_seed)
        output_sequence = transformer_model.forward(input_sequence)
        output_vector = extract_last_vector(output_sequence)
        permutation = generate_random_permutation(input_seed)
        permuted_vector = permute_vector(output_vector, permutation)
        distance = compute_euclidean_distance(permuted_vector, target_vector)
        if distance < threshold:
            send_valid_nonce(nonce, node_public_key)
            valid_nonces.append(nonce)
        if timer.check() > race_duration:
            break
    return valid_nonces
```

```
def validate_proofs(node_public_key, submitted_nonces,
blockchain_state_at_race):
    """
    Validates submitted nonces based on the blockchain state at the race time.
    Using a race seed from the blockchain state and the node's public key,
    the function checks if each submitted nonce generates apprpriate vector.

    Args:
        node_public_key (str): Node's public key.
        submitted_nonces (list): Nonces to validate.
        blockchain_state_at_race (str): Blockchain state at race time.

    Returns:
        list: Validated nonces.
    """
    race_seed = generate_race_seed(blockchain_state_at_race)
    transformer_model = initialize_transformer(race_seed)
    node_seed = generate_node_seed(node_public_key)
    target_vector = generate_target_vector(race_seed)
    valid_proofs = []
    for nonce in submitted_nonces:
        input_seed = combine_seeds(node_seed, race_seed, nonce)
        input_sequence = generate_input_sequence(input_seed)
        output_sequence = transformer_model.forward(input_sequence)
        output_vector = extract_last_vector(output_sequence)
        permutation = generate_random_permutation(input_seed)
        permuted_vector = permute_vector(output_vector, permutation)
        distance = compute_euclidean_distance(permuted_vector, target_vector)
        if distance < threshold:
            valid_proofs.append(nonce)
    return valid_proofs
```

Appendix B

Random number generation

A key element of the Race is generating the random number that seeds the computational task. This random number must be truly random and immune to manipulation by any participant. Ensuring the integrity and fairness of this process is crucial, as it forms the basis for subsequent computations and consensus mechanisms in the network. To achieve this, the system employs a combination of classic cryptographic algorithms alongside advanced techniques like Threshold Cryptography with Synchronization and the Commit–Reveal with Timed Reveal mechanism. These methods are integrated to guarantee randomness, security, and decentralization at every step of the process.

- **Threshold Cryptography with Synchronization.** In the decentralized AI network, the random number is generated collaboratively by multiple nodes using a Threshold Cryptography scheme. The secret (random number seed) is split into multiple fragments and distributed among a group of nodes, each holding a share of the secret. To reconstruct the final random number, a predefined threshold of nodes must combine their shares. This threshold system ensures that no individual node influences the outcome or generates the number independently. Synchronization plays a critical role in ensuring that these nodes cooperate in real time to reconstruct the number. By synchronizing their actions, nodes prevent any delay, coordination failure, or manipulation of the process. This mechanism significantly strengthens the network's defense against malicious actors by distributing the responsibility for number generation and preventing any single point of failure.
- **Commit–Reveal with Timed Reveal.** In addition to Threshold Cryptography, the system incorporates a Commit–Reveal with Timed Reveal scheme to further secure the random number generation process. During the commit phase, each node submits a cryptographically hashed version of its intended contribution to the random number. These commitments are locked and stored within the system, preventing any node from altering its input after viewing the contributions of others. The reveal phase occurs after all commitments are received. At this stage, each node reveals its original value, which is verified against the earlier commitment. The timed reveal aspect enforces a specific time window for the reveal phase, ensuring that all participants disclose their inputs simultaneously. This prevents any participant from withholding or manipulating their commitment, safeguarding the fairness and transparency of the process.

These methods ensure that the final random number is the product of contributions from multiple participants, each with voting weight proportional to their computational effort in the previous race. By tying voting weight to prior contributions, the system maintains a balance of fairness and computational influence. This mechanism guarantees that no single participant can predict or influence the outcome, ensuring the random number remains unbiased and unpredictable.

At the end of each cycle, the group of participants with the highest accumulated voting weights must repeat this random number generation procedure just before creating the next k*n-block in the blockchain. The result of this block generation is a new random number, which will be used to seed the next race. This continual regeneration of random numbers ensures that each race is independent and unpredictable, further reinforcing the system's integrity and the equitable distribution of computational influence within the network.

Appendix C

Aspect	Proof-of-Stake	Proof-of-Work	Proposed Novel Consensus
Voting Weight Basis	Based on the amount of cryptocurrency staked	Based on computational power and work performed	Based on computational work during the Race
Task Focus	Mainly securing the network	Computational work dedicated solely to network security	Meaningful tasks such as AI model training and inference
Resource Allocation	Tied to the amount of staked capital	All resources are directed to computational tasks for network security	Resources allocated efficiently for both network security and productive tasks
Efficiency in Utilization	Efficient in terms of energy but less so in resource allocation	Inefficient, as 100% of computational work goes towards non-productive tasks	Optimized efficiency by directing resources toward useful tasks
Energy Consumption	Low energy consumption compared to PoW	High energy consumption	Reduced energy consumption by avoiding wasteful computational tasks
Reward Distribution	Mostly directed towards capital holders and securing the network	Entirely consumed by validating non-productive computations	~100% allocated to productive, meaningful tasks
Incentives for Productive Work	Incentives are not strongly tied to computational contributions	No incentives for meaningful work. 0% are directed to productive tasks	Strongly incentivizes participants to contribute to AI advancements and productive tasks

Appendix D

Hardware Specifications

Criteria: Nvidia GPUs belonging to generations after Tesla, with a minimum of 16 GB VRAM per GPU.

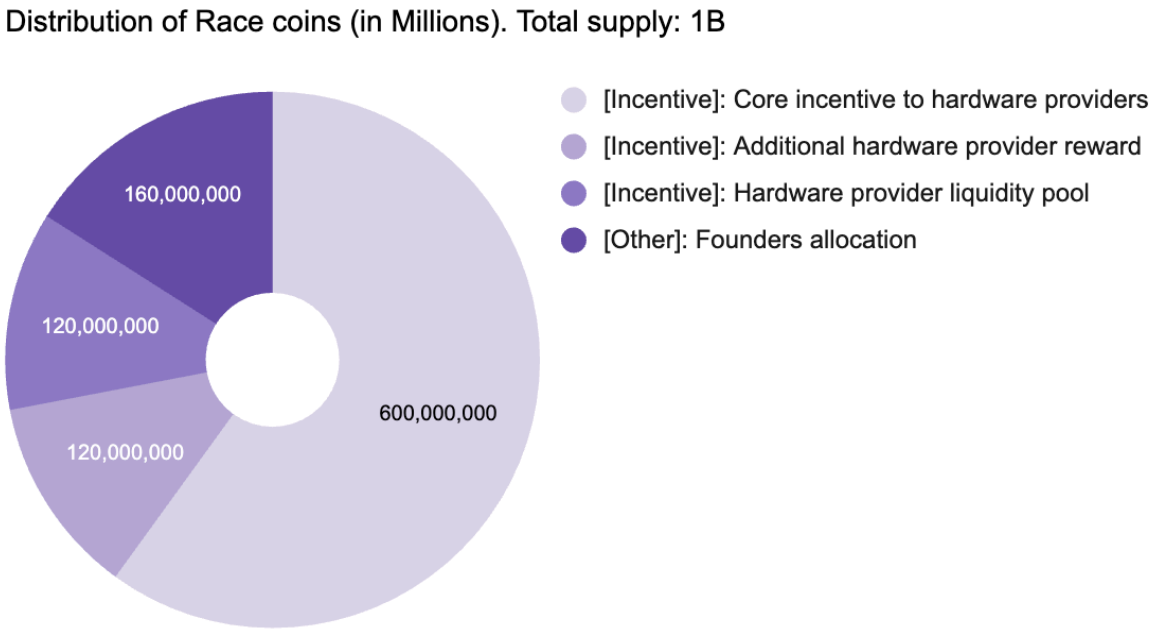
NVIDIA GPU	Release Date	VRAM	Architecture	Generation
RTX 3090	September 2020	24 GB GDDR6X	Ampere	RTX 30 Series
A100	May 2020	40 GB or 80 GB HBM2e	Ampere	A100
RTX A6000	December 2020	48 GB GDDR6	Ampere	RTX 30 Series
A30	2021	24 GB HBM2	Ampere	A30
A10	2021	24 GB GDDR6	Ampere	A10
A40	2021	48 GB GDDR6	Ampere	A40
L40	2022	48 GB GDDR6	Ada Lovelace	L-Series (Data Center)
H100	May 2022	80 GB HBM3	Hopper	H100 (Data Center)
RTX 4090	October 2022	24 GB GDDR6X	Ada Lovelace	RTX 40 Series
RTX 6000 Ada Gen	December 2022	48 GB GDDR6	Ada Lovelace	RTX 40 Series
L4	March 2023	24 GB GDDR6	Ada Lovelace	L-Series (Data Center)
H200	2024	141GB of HBM3e	Hopper	H100 (Data Center)

Appendix E

Tokenomics

The total supply of Race Coins is fixed at 1 billion, allocated to incentivize hardware providers, support network development, and ensure fair compensation for contributors. Below is the detailed breakdown of the allocation. The distribution is divided into two primary categories:

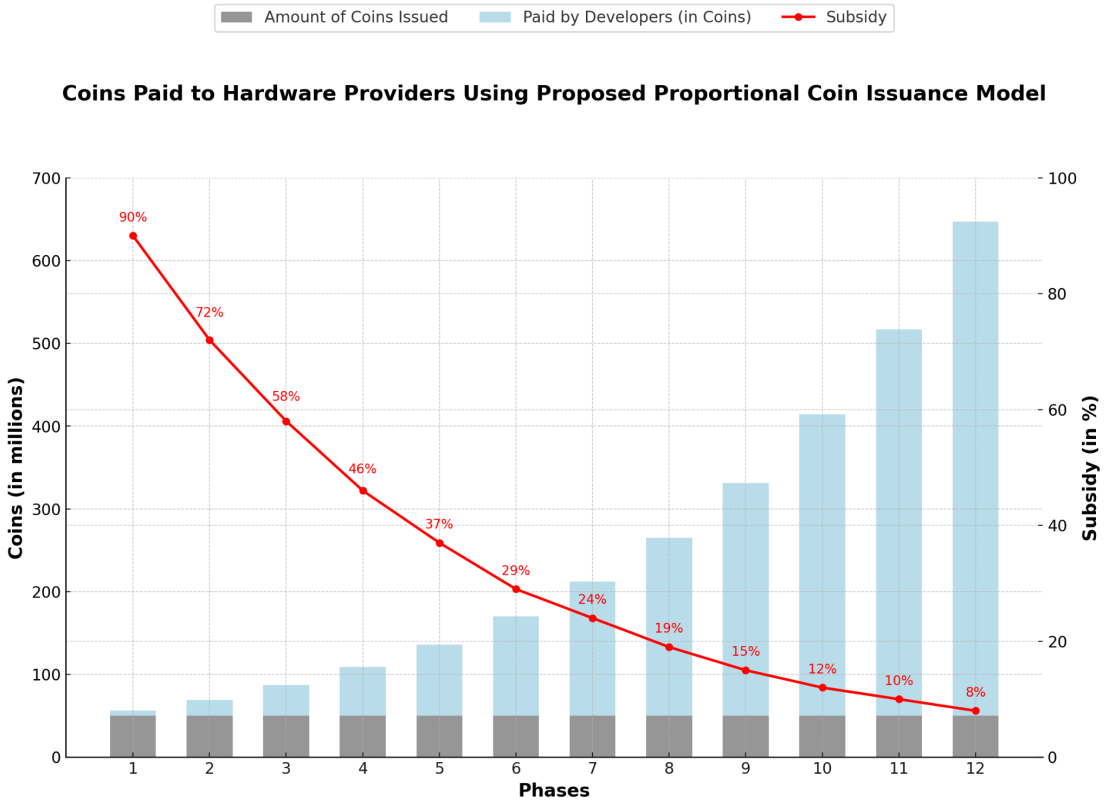
1. **Incentives to Hardware Providers** (840 million Race Coins, 84% of total supply).
2. **Other: Founders allocation** (160 million Race Coins, 16% of total supply) is a portion reserved for the founding team as compensation for their efforts in developing and launching the network.



Incentives to Hardware Providers consist of the following:

1. **Core incentive to hardware providers** (600 million Race Coins, 60% of total supply) is designed to reward hardware providers for contributing computational power to the decentralized network. Similar to Bitcoin mining, hardware providers directly receive newly minted coins proportional to their computational contribution.

   To effectively distribute these incentives, the Proportional Coin Issuance Model is introduced. Under the structured token-distribution mechanism, the issuance of Race Coins occurs gradually across predefined phases. Each phase issues a fixed batch of 50 million coins, while progressively reducing the network-funded subsidy for hardware providers. Specifically, subsidies begin at 90% in Phase 1 and systematically decrease down to 8% by Phase 12, following a defined 1/5-life reduction model. Although the exact timing between phases remains undetermined, the issuance strategy is to align each phase roughly with the network's annual growth milestone (~1-year interval).



This Proportional Coin Issuance Model is specifically designed to balance incentives for early-stage adoption with long-term scalability of the network. By progressively aligning token issuance with network growth, the model ensures fair compensation for hardware providers.

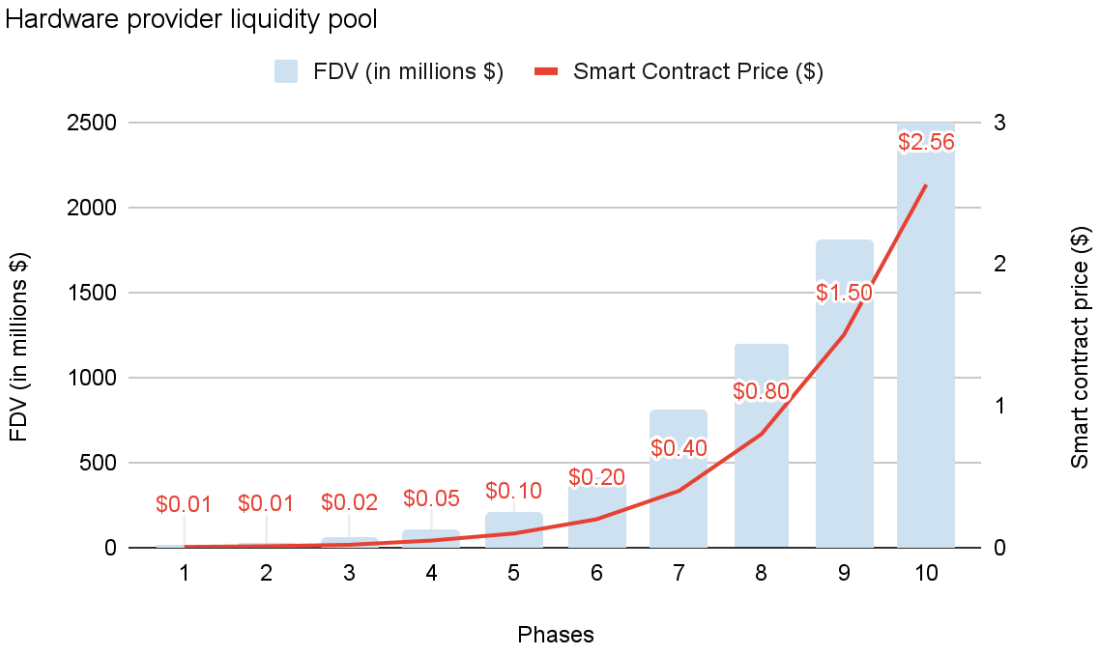
In the network's early stages, demand for inference service may be limited due to slower adoption rates by developers utilizing AI models (Developers). Without incentive, hardware providers might find it economically challenging to contribute computational power consistently while demands is low. To address this, **the subsidy mechanism** acts as a stabilizing force, guaranteeing fair compensation on all stages of network development. This ensures the network maintains adequate computational capacity, thereby supporting early adopters and core functionality. As adoption accelerates and Developer demand for inference increases, hardware providers begin earning more market-driven rewards. At this stage, the need for network-funded subsidies diminishes. The gradual subsidy reduction ensures a smooth transition from subsidy reliance toward self-sustaining, market-driven reward system.

2. **Additional Hardware Provider Reward** (120 million Race Coins, 12% of total supply) is targeted at recognizing and additionally rewarding early, high-performing hardware providers. Specifically, this 12% allocation is divided among the first top miners who each committed at least 1,000 A100 GPUs and sustained their contribution for 4 years. This structured reward mechanism ensures long-term stability and incentivizes sustained participation from high-performance miners. By gradually distributing the reward over four years, the system prevents short-term exploitation and encourages miners to maintain their contribution consistently. Additionally, the mechanism allows for dynamic replacement - if a top miner exits,

the next eligible miner inherits the remaining rewards, ensuring continuity without disrupting the network.

3. **Hardware provider liquidity pool** (120 million Race Coins, 12% of total supply) is designed to provide early liquidity for Hardware Providers before Race Coin is adopted by exchanges and P2P trading volume reaches levels comparable to similar projects. This pool will be distributed through a preprogrammed sale mechanism that allows Developers and token holders to buy Race coins at any time.

   Buyer can transfer USDT, Ethereum, or Bitcoin to a designated account on the network, automatically minting Race Coins via the native bridge. The price follows a bonding curve, meaning it increases exponentially as more coins are minted. Meanwhile, Hardware Providers will be able to send their mined Race Coins to a dedicated smart contract on Race network and receive USDT, Ethereum, or Bitcoin from the pool at the latest achieved price, ensuring early liquidity.



For example, price changes can be as follows: The coins will be released in phases. The release of the next phase is contingent on the complete sale of the preceding phase. The first phase releases 12 million coins at a starting price of $0.005. Once all coins in the first phase are sold, the second phase is released at a higher price, increasing to $0.01, followed by subsequent phases with progressively higher prices. This “sell-out-before-next-phase” approach incentivizes early investment by allowing early participants to acquire Race Coins at the lowest price point. As each phase sells out, the scarcity of available coins drives demand, rewarding participants who act early.

Appendix F

Tokenomics. Reasoning behind the Proportional Coin Issuance Model

The token issuance model is designed to achieve two primary objectives: fostering network growth during its early stages and ensuring long-term sustainability. Drawing on principles from Proof-of-Work systems, the model integrates utility-token mechanics to create a predictable and efficient framework. This Appendix explains the principles behind token issuance, the logic guiding its design, and the safeguards that prevent exploitation.

Tokens play a fundamental role in supporting the network by subsidizing early growth and ensuring long-term collaboration between two key participants: hardware providers and Developers.

In the network's initial phase, when the number of inference requests is not sufficiently high, the network will be subsidized for incentivising participation:

- Subsidies motivate hardware providers to contribute the computational power required to support network operations.
- Subsidies reduce costs, enabling developers to deploy and run AI services on the network at significantly lower expenses compared to centralized alternatives.

These incentives help establish a robust foundation, encouraging adoption and ensuring sufficient activity to drive the network's growth. While hardware providers compete to deliver services at the lowest effective cost (calculated as operational expenses minus subsidies), Developers benefit from predictable costs, even as subsidies decline, allowing them to plan operations and scale effectively.

Race Coin's issuance model is built on the best practices from the market while adding its own innovations for scalability and efficiency. Inspired by Bitcoin and Ethereum, Race Coin's issuance model is based on a hybrid approach that adapts dynamically to the network's demand and usage. To understand this approach, it's important to understand how Bitcoin and Etherium have structured their token distribution to incentivize participation and ensure long-term sustainability. In the beginning, more tokens are usually issued to incentivize participation:

- In **Bitcoin**, about 12.51% of Bitcoin's total supply was distributed in its first year (50 BTC per block). [16] This reward undergoes a halving approximately every four years, reducing the issuance rate by half. The first halving in 2012 reduced the reward to 25 BTC, followed by 12.5 BTC in 2016, 6.25 BTC in 2020, and 3.125 BTC in 2024. This process continues until the total supply of 21 million BTC is distributed. During this issuance period, miners are incentivized by two mechanisms:
    - Transaction Fees: Paid by users for transactions.
    - Block Rewards: In the form of newly minted BTC.

  Together, these incentives ensure sufficient motivation for miners to participate actively in securing the network. Gradually, all tokens are distributed, and it is assumed that no additional investment will be required, as transaction fees will be sufficient for the network's operations to continue without the need for additional injections.
- **Ethereum** followed a different model. Initially, the annual issuance rate under Proof-of-Work was approximately 6.7% of the total supply. [17] With the transition to Proof-of-Stake through "The Merge," the issuance rate dropped significantly to around 0.52%. [17] Although Ethereum's approach differs from Bitcoin's time-based halving, both systems gradually reduce issuance as the network matures.

Race's dynamic supply-driven halving
Unlike Bitcoin's time-bound halving or Ethereum's rate adjustments, Race adopts a Hybrid Proportional Coin issuance model with a minimum independent Coin issuance (set at 5% of the total supply for the first year and will decrease by half each subsequent year) ties directly to network activity.

The total supply of the Race Coin is 1,000,000,000. Subsidies will be issued by batches (1 batch = 50M Race Coins, equivalent to 5% of the total supply). Each time 5% of the total token supply is issued, the subsidy decreases by 20%. This balances incentives with network demand, ensuring token issuance aligns with usage and avoids unnecessary inflation.

#	Race Coins	1/5-life reduction of subsidies	Fees paid by developers in Race Coins	Total paid to hardware providers in Race Coins
1	50,000,000	90.00%	5,555,556	55,555,556
2	50,000,000	72.00%	19,444,444	69,444,444
3	50,000,000	57.60%	36,805,556	86,805,556
4	50,000,000	46.08%	58,506,944	108,506,944
5	50,000,000	36.86%	85,633,681	135,633,681
6	50,000,000	29.49%	119,542,101	169,542,101
7	50,000,000	23.59%	161,927,626	211,927,626
8	50,000,000	18.87%	214,909,532	264,909,532
9	50,000,000	15.10%	281,136,915	331,136,915
10	50,000,000	12.08%	363,921,144	413,921,144
11	50,000,000	9.66%	467,401,430	517,401,430
12	50,000,000	7.73%	596,751,788	646,751,788
	600,000,000			

Minimum independent Coin issuance
Hybrid Proportional Coin Issuance Model ensures that a minimum number of new coins are created each year, regardless of demand.

How it works:

- At the beginning of each year, the system sets a minimum daily issuance rate, which is calculated based on a percentage of the total supply. This ensures that a certain number of coins enter circulation every day, regardless of demand.
    - In the first year, this minimum is 5% of the total supply. 5% of 1 billion = 50 million coins. Daily issuance: 50,000,000 ÷ 365 = 136,986 coins per day
    - In the second year, it reduces to 2.5% of 1 billion = 25 million coins. Daily issuance: 25,000,000 ÷ 365 = 68,493 coins per day.
    - In the third year, it further drops to 1.25% of 1 billion = 12.5 million coins. Daily issuance: 12,500,000 ÷ 365 = 34,247 coins per day
    - Etc.

- Hardware providers earn rewards based on the amount of work they have done. However, if demand is so low that the set minimum coin issuance is not fully utilized, this minimum issuance will still be distributed among those who have contributed work, in proportion to their efforts.

The tokenomics model provides clear and predictable costs for developers, making it easier for them to plan long-term participation:

- **Early stages**. Developers can expect computational costs to be up to ten times cheaper than market rates due to the 90% subsidy provided during the initial phases. For example, if a developer spends X tokens on a request, hardware providers may receive up to 9X in rewards as part of the subsidy.
- **Gradual adjustment.** As subsidies decline with each halving, developers experience steady and predictable cost increases. This transparency ensures that developers can budget for growth without being affected by sudden or unpredictable changes.

To prevent exploitation of the subsidy structure, the model incorporates randomized task distribution. Tasks are distributed randomly across hardware providers, ensuring no participant can disproportionately benefit from artificially generated requests. Developers have no control over which provider processes their tasks.

Also, as subsidies decrease, market efficiency is expected to rise, driven by advancements in hardware performance. Just as the Bitcoin network saw the development of highly efficient ASICs, the network anticipates similar innovations, reducing the cost of computational power provision and ensuring sustainability as rewards shift entirely to transaction fees.

Appendix G

Tokenomics. Hypothetical* Competitive Analysis

**This model is a hypothesis, not a promise, designed to explore the potential of balancing scalability and fairness in a decentralized network. If the Race Coin network achieves adoption rates comparable to competitors like Bittensor (Year 3) or OpenAI (Year 6), hardware providers stand to benefit significantly. However, these outcomes depend on network growth and are subject to market dynamics.*

Year 3 Comparison with Bittensor

Competitor Analysis:

Scenario 1 - Bittensor by year 3

Market Cap (in millions)		FDV (in millions)	
Bittensor	Race Coin	Bittensor	Race Coin
$3,100	$410	$8,820	$892
	-655% Undervalued		-888% Undervalued

Difference in Market Cap: $3.1B (Bittensor) vs. $410M (Race Coin).
Race Coin hypothetically undervalued by 655%.

Difference in FDV: $8.8B (Bittensor) vs. $892M (Race Coin).
Significant room for Race Coin valuation growth.

Data inputs and assumptions for this analysis:

- **Bittensor metrics (December 2024)**
  - Approximate request volume (in millions AI-tokens): 7.8M
  - Market Cap: $3.1B
  - FDV: $8.8B
  - Market Price per 1M AI-tokens (2024): $10
- **Race assumptions (used in calculations below)**
  - Approximate discount expected by Developers compared to market price: 57.6
  - Fees paid by Developers in Race Coins (Year 3, not cumulative): 36.8M Race Coins
  - Race Coins issued by Year 3:3 batches x 50M + 60M (Hardware provider liquidity pool) + 90M (First three years of additional hardware provider reward) + 160 Founders allocation = 460M

Based on these assumptions the following mathematical calculations are performed:

1. **Adjusted market price per 1M tokens**
   Market price per 1M tokens (2024) — Discount $= \$10 - 57.6\% = \$4.2$

2. **Cost per 1M requests in Race Coins to serve request volume similar to Bittensor**

$$\frac{Fees\ Paid\ by\ Developers\ (Year\ 3)}{Request\ Volume\ (M\ Tokens)} = \frac{36.8M}{7.8M} \approx 4.7\ Race\ Coins\ per\ 1M\ requests$$

3. **Price per Race Coin (based on calculations above)**

$$\frac{Market\ price\ per\ 1M\ tokens\ (2024)\ reduced\ by\ the\ expected\ discount}{Race\ Coins\ per\ 1M\ requests} = \frac{\$4.2}{4.7} \approx \$0.89\ per\ Race\ Coin$$

4. **Market valuation calculation**

$Price\ per\ Race\ Coin \times Total\ Coins\ issued\ by\ Year\ 3$

$\$0.89 \times 460.000.000 \approx \$410M$

5. **Fully Diluted Valuation (FDV) calculation**

$\$0.892 \times 1.000.000.000 \approx \$892M$

The calculations highlight that Race Coin could deliver comparable network functionality with significantly lower operational costs.

Race network in Year 6 vs. Major centralized market player

Scenario 2 - OpenAI by year 6

Market Cap (in millions)		FDV (in millions)	
OpenAI	Race Coin	OpenAI	Race Coin
$157,000	$13,967	$157,000	$21,823
	-1024% Undervalued		-619% Undervalued

Difference in Market Cap: $157B (OpenAI) vs. $13.9B (Race Coin).
Race Coin undervalued by 1024%

Difference in FDV: $157B (OpenAI) vs. $21.8B (Race Coin).
Illustrates network's scalability potential.

Data inputs and assumptions for this analysis:

- **OpenAI Metrics (December 2024)**
  - Approximate request volume (in millions AI-tokens): 370M
  - Market Cap and FDV: $157B
  - Market price per 1M AI-tokens (2024): $10
- **Race assumptions (used in calculations below)**
  - Approximate discount expected by developers in Year 6: 30%
  - Fees paid by Developers in Race Coins (Year 6, not cumulative): 119M
  - Race Coins issued by Year 6: 6 batches x 50M + 60M (Hardware provider liquidity pool) + 120M (Additional hardware provider reward) + 160 Founders allocation = 640M

Based on these assumptions the following mathematical calculations are performed:

1. **Adjusted market price per 1M tokens**

Market price per 1M tokens (2024) — Discount $= \$10 - 30\% = \$7$

2. Cost per 1M requests in Race Coins to serve request volume similar to OpenAI

$$\frac{Fees\ Paid\ by\ Developers\ (Year\ 6)}{Request\ Volume\ (M\ Tokens)} = \frac{119M}{370M} \approx 0.32\ Race\ coins\ per\ 1M\ requests$$

3. Price per Race Coin (based on calculations above)

$$\frac{Market\ price\ per\ 1M\ tokens\ (2024)\ reduced\ by\ the\ expected\ discount}{Race\ coins\ per\ 1M\ requests} = \frac{\$7}{0.32} = \$21.82\ per\ Race\ Coin$$

4. Market valuation calculation

$$Price\ per\ Race\ Coin \times Total\ Coins\ Issued\ by\ Year\ 6 = \$21.823 \times 640\,000\,000 \approx \$13,967M$$

5. Fully diluted valuation (FDV) calculation

$$Price\ per\ Race\ Coin \times Total\ Supply\ of\ Race\ Coins = 21.823 \times 1,000,000,000 = 21,823M$$

The calculations highlight that Race Coin could deliver comparable network functionality with significantly lower operational costs due to its decentralized approach.

Appendix H

Tokenomics. Financial benefits for developers and hardware providers

For hardware providers:

1. Early growth and high reward potential:
   a. Race Coins are in their early lifecycle, with significant network growth ahead.
   b. Hardware providers benefit from low valuation, accumulating more coins before demand scales.
   c. As inference requests grow, rewards will increase, tied directly to network adoption.
   d. When Race Coins' FDV reaches $8.8B (Bittensor's level) by Year 3, early participants could see a 100x increase in earned coin value.
2. Our Tokenomics is developed with mining providers interests at its core:
   a. Direct token rewards for computational contribution:
      i. 84% of total Race Coins (840M) allocated for hardware provider incentives.
      ii. Compensation is based on actual computational work, ensuring fair rewards.
   b. Early liquidity support:
      i. 120M Race Coins set aside for an early liquidity pool.
      ii. Hardware providers can exchange mined tokens for stable assets (USDT).
   c. 120M Race Coins set aside as an additional reward for the TOP hardware providers committing large-scale GPU power.
   d. Gradual subsidy reductions prevent over-inflation and ensure steady demand for Race Coins.
3. Optimal use of mining computational power and miners contribution to AI future:
   a. Race doesn't use miners computational power (and electricity) in the absence of tasks. Mining computational power can be used on other platforms while there is nothing to compute at Race.
   b. Mining of Race Coins is a way for individual hardware owners to contribute to and benefit from AI computational markets
   c. Race Coins help to diversify revenue streams with less speculative, therefore more stable demand for compute power
   d. GPU crypto mining clusters have compute power comparable with the top AI companies and cloud providers. But a lot of crypto resources are used for token security vs useful computation. Race Coins are optimised for computational outcome (not compromising security) the majority of mining power contributes directly to AI future.

For developers:

1. Lower AI computation costs:
   a. Initial subsidies (up to 90%) significantly reduce inference and training expenses. For example, if a developer spends X tokens on a request, hardware providers may receive up to 9X in rewards as part of the subsidy.
   b. Costs remain lower than centralized providers like AWS, Azure, and OpenAI.
2. Predictable and gradual cost adjustments:
   a. Well-structured subsidy reductions allow for long-term financial planning.
   b. Avoids sudden price hikes seen in centralized cloud services.
3. Developers who fund or contribute to model training earn a share of future inference revenue.
4. Early participation rewards:
   a. Developers using the network in its early phases benefit from lower-cost computational resources.
   b. Increased token value over time can enhance early adopters' financial returns.
5. Easy to use API with no DevOps (similar platform.openai.com)
6. Mitigates risk of big centralized AI platforms, their censorship and anti-competitive practices.

Appendix I

Tokenomics. Risks and Considerations

While the proposed issuance model offers significant earning potential for hardware providers and investors, particularly in early stages, its success depends on achieving adoption milestones, managing token value stability, and fostering trust through transparent governance.

Hardware providers must carefully weigh the potential benefits of early participation against the following risks:

1. Market volatility risk - the hypothesis relies on achieving significant developer adoption and sustaining demand for inference services. Fluctuations in the token's market value may reduce the real-world value of rewards earned by hardware providers and unpredictable earnings may deter long-term investments in infrastructure. Transparent communication and governance can reduce speculation and foster token stability.
2. Regulatory uncertainty risk - tokens issued as rewards could face regulatory scrutiny, especially in jurisdictions with strict cryptocurrency laws. The model assumes compliance with decentralized network principles and thus, no classification as a security in the United States and other jurisdictions with similar approaches to crypto asset regulation. As such, all figures and growth projections are theoretical and should not be construed as investment promises. Legal challenges may impact token issuance and create unforeseen compliance burdens for providers. Adherence to decentralized principles and proactive monitoring of legal frameworks may, but is not guaranteed to, minimize risk exposure.
3. Subsidy reduction over time - gradual reduction in developer subsidies (from 90% in Phase 1 to 8% in Phase 12) could discourage developer participation and lower developer engagement may reduce network usage, impacting proportional rewards for hardware providers. Adjusting subsidy reduction rates based on adoption metrics could align reward structures with actual growth.
4. Technological and market competition risk - centralized providers or competing networks may develop more efficient systems, reducing demand for decentralized alternatives. Providers could see reduced utilization of their infrastructure, however, differentiation through decentralization, transparency, and competitive rewards is key and can minimize risk exposure.

Appendix J

AI inference pricing model

AI inference pricing mechanism is built around a standardized unit of compute, allowing participants to estimate computational expenses in a universally accepted format. The system dynamically adjusts pricing through a decentralized voting mechanism, ensuring fairness and adaptability based on real-time supply and demand.

Key concepts of the pricing model

1. Unit of compute is a standardized measure representing the computational effort required for AI operations (inference and model training). It's used to create a uniform pricing structure.
2. Prices can be set in multiple denominations (different scales of the network's native currency):
   - icoin
   - uicoin (micro-coins)
   - micoin (milli-coins)
   - nicoin (nano-coins)
3. Market-driven pricing is determined by how many blockchain coins correspond to one unit of compute. To achieve a fair market price:
   a. Each hardware provider submits a vote indicating how many coins per unit of compute they propose.
   b. The weighted median of these submissions is determined at the end of each epoch (when validators are rotated).
   c. That weighted median becomes the common price per unit of compute for the next epoch.
4. If a hardware provider never votes, the most recently used epoch price (or the genesis parameter for the very first epoch) is used as their default vote.

Automated API for Price Management

Hardware providers can interact with the pricing system using an API that allows them to:

- Set or update price proposals
- Check their current price vote
- Query the latest global unit of compute price and model pricing
- Register new AI models

Dynamic Pricing for AI Models

Each AI model consumes a specific number of unit of compute per processed token. The pricing model ensures transparency by:

1. Listing models with their unit of compute price per token.
2. Setting a price per token derived from the unit of compute price.
3. Allowing new models to be proposed and voted on before they are approved for the network.

Decentralized Model Registration Process

1. Developers submit a model proposal with its estimated number of units of compute required for each token of that model.
2. The governance system, powered by the CosmoSDK, ensures that:
   a. A deposit is made to prevent spam.
   b. The proposal is reviewed by participants.
   c. Votes are cast to approve or reject the model.
3. Once approved, the model is listed in the pricing structure and available for computational tasks.

Appendix K

Hardware providers' compensation model for training new Large Language Models

The decentralized AI infrastructure is built on a foundational commitment to open-source development and accessibility. Advancing AI should not be controlled by centralized entities only. Instead, it must remain collaborative, transparent, and widely available to developers and users worldwide. To uphold this commitment, the network guarantees that all models trained using its resources will remain open-source, ensuring that AI innovation remains decentralized and equitable. The network actively commits to continuously contributing to open-source AI development, setting it apart from centralized entities like Meta, which do not provide such guarantees. This commitment fosters motivation among contributors and ensures that innovation remains in the hands of a global community rather than a select few.

However, training new Large Language Models is one of the most computationally intensive processes in AI. Fair and sustainable compensation for hardware providers is essential to keeping the decentralized AI network competitive and encouraging long-term participation.

To achieve this, the network adopts a Network-Financed approach, where 20% of all inference revenue (including transaction fees and mining rewards) will be allocated to fund the training of new models. This allocation will specifically cover the cost of the unit-of-compute used during training, ensuring that resources are efficiently utilized to support model development. Additionally, a portion of this 20% can be allocated as grants, awarded through a voting process, to support the most promising and iinovative traning procedures proposed by community. This mechanism ensures that the network continues to support large-scale AI advancements, even after initial subsidies phase out. Moreover, as the network grows, this approach allows for the collection of substantial funding, making it feasible to train massive AI models that compete with centralized alternatives.

The network will implement a consensus-based governance mechanism to maintain flexibility in adjusting the revenue allocation percentage. If new models significantly boost network adoption, consensus may choose to maintain or increase the 20% allocation. If the allocation impacts network competitiveness, consensus can reduce the percentage, ensuring a balance between affordability and innovation. This mechanism will only become available after Year 5, ensuring the network has demonstrated sufficient model training capabilities before modifications are allowed.

The decision on how to conduct the next training procedure follows a clear yet flexible process to ensure the most effective training approaches receive support. Rather than focusing solely on which model to train, the emphasis is on refining the training methodology. Contributors, including hardware providers and developers, can propose code changes (such as pull requests in a repository) that introduce new training strategies. These proposals undergo thorough discussion, with extensive presentations and debates to assess their feasibility and effectiveness.

Once a training approach gains consensus, it is first approved as a code update. Following this, the same participant can formally propose a full training experiment, detailing the parameters, dataset, and allocated funding. This proposal then enters a governance voting phase, where it is likely to undergo multiple iterations, including rejections and modifications before reaching final approval. The network ensures that the best AI research and development teams worldwide can propose and experiment with promising training techniques, even if they lack their own hardware. This ensures that innovative research is not limited by computational constraints while maintaining full transparency (everyone can still access and evaluate R&D results, fostering continuous innovation). Priority is given to training experiments that demonstrate strong potential to maximize impact, ensuring resources are directed toward AI advancements that offer the greatest value. This iterative process fosters innovation, aligns the network’s

growth with user needs, and ensures that training methodologies continue to evolve in the most effective way possible.

Why the Decentralized AI Training Fund?

Various approaches were considered for financing LLM training. The Decentralized AI Training Fund was chosen due to its sustainability, fairness, and ability to encourage meaningful participation from hardware providers.

Financing Model	Advantages	Challenges
Decentralized AI Training Fund (chosen)	● Ensures stable, long-term funding; ● Predictable rewards for hardware providers; ● Scalable as network revenue grows	● Requires governance to adjust revenue allocation
New Coin Issuance	● Immediate incentives for providers; Allows early growth	● May lead to inflation and reduced token value; Unclear incentive alignment
Shareholding Structure	● Participants receive compensation through additional fees paid by developers, along with a proportional share of the network reward. ● Hardware providers receive a stake in the model and are compensated whenever it is used. ● Creates long-term incentives to support training.	● Developers using trained models may pay an additional fee. ● Additional usage fees may be at a disadvantage against free alternatives like LLaMa (funded by Meta). ● Network may subsidize the cost, though this could result in slightly lower earnings for hardware providers. ● Technical challenges related to the rights for the model weights, which could lead to security threats and potential attacks targeting model ownership.

The Decentralized AI Training Fund ensures long-term funding for LLM training, balancing sustainability with competitive pricing. By allocating 20% of inference revenue to model training, the system secures billions in funding while allowing for governance-based adjustments in later years. This approach benefits hardware providers and developers fostering a scalable, decentralized AI ecosystem that can compete with centralized cloud solutions while maintaining fairness and efficiency.

Participant-financed training relies on direct funding from hardware providers and developers, creating a more demand-driven approach. While this model requires users to actively contribute to financing, potentially limiting participation, it remains a viable option for those who choose to fund training independently. Additionally, while there is a risk of model duplication and free redistribution, this approach offers flexibility for individuals or organizations willing to invest directly in model development.

Appendix L

The DiLoCo Mechanism

In classical distributed learning, nodes typically synchronize after every training step, leading to enormous communication overhead and waiting times as models grow larger. DiLoCo [18] challenges this approach by introducing a different synchronization strategy. Instead of continuous parameter exchange, DiLoCo synchronizes the model's parameters only periodically (e.g., every 1000 training steps).

What enables DiLoCo to synchronize less frequently is its bi-level optimization process, building upon foundational work in federated learning [21][22]. Each node performs multiple iterations independently using optimizers such as AdamW, while the outer optimization loop, inspired by Federative Averaging, synchronizes local models every N steps. This ensures efficient use of local compute resources. The outer optimization loop, unique to DiLoCo, introduces advanced techniques like Nesterov momentum to aggregate updates across nodes periodically. By carefully pairing these optimizers, DiLoCo achieves both stability in global updates and flexibility in local computation.

The outcome is a learning environment capable of operating effectively across geo-distributed networks. By drastically lowering communication overhead, DiLoCo enables more efficient large-scale training than traditional distributed approaches which require every step synchronization. DiLoCo's scaling laws [25] show predictable performance improvements as model sizes grow, making it particularly suitable for training foundation models. Practical implementations of DiLoCo can train 30-50 billion parameter models using configurations of 8xH100 GPUs servers.

Transition from Centralized to On-Chain Management

Originally, DiLoCo implementations have relied on a centralized node that managed the synchronization schedule, rendezvous, and assigned ranks to the nodes. This structure introduced a single point of failure and potential trust concerns. To resolve these issues, our approach replaces the centralized node with a decentralized on-chain management framework.
Critical processes such as rendezvous, rank assignment, and synchronization coordination are now handled through distributed blockchain mechanisms. This ensures that no single entity controls the training process, reinforcing trustlessness and resilience to probable node failure/disconnect.
By incorporating on-chain management, the training process becomes more democratic and transparent, preventing any single authority from unilaterally affecting outcomes, manipulating global parameters or disrupting the whole training process.

Proof-of-Learning and Trustless Collaboration

In a decentralized training environment, all participants are inherently untrusted. This fundamental assumption drives our approach to validation. We need mechanisms that allow participants to verify others are not misrepresenting their work or manipulating the learning process.

Our approach incorporates elements inspired by “Proof of Learning” [19], ensuring that participants who falsify training will eventually be detected and penalized. The system employs probabilistic validation whereby participants are subject to verification at pseudo-randomly selected intervals. This approach strikes a balance between comprehensive oversight and computational efficiency.

State Preservation

At predetermined, pseudo-randomly selected steps during training, each participating node preserves critical artifacts to verify training steps: previous weights, current weights, and optimizer states (including momentum terms and adaptive learning rates). These preserved states serve as evidence of legitimate training activities and create a verifiable proof of the learning process.

Participants commit hashes of their preserved states to the blockchain. These hash commitments provide timestamped, tamper-evident records of training progress while requiring minimal on-chain storage. The actual training artifacts remain off-chain but have to be provided on demand during validation procedures.

Validation Process

The validation process operates through randomly selected validators drawn from participants not currently engaged in the model training process. These validators periodically request the underlying data from training nodes, verify that the hash matches the on-chain commitment, and confirm that weight changes represent legitimate training iterations based on the provided data. Should discrepancies arise—for instance, if a node attempts to submit falsified weights—validators flag these issues on-chain, preserving network integrity.

Crucial to our approach is the understanding that validators themselves cannot be inherently trusted. To address this, validation results are permanently recorded on-chain, allowing any participant to independently verify these assessments at any time. We implement a multi-layered verification system where validators are themselves subject to oversight through honeypot traps (intentionally flawed submissions designed to test validator diligence) inspired by [19] and re-validation by other validators.
This mechanism ensures a trustless, verifiable training environment. Malicious actors attempting to fake progress or insert incorrect weights would be promptly caught, upholding data integrity and fostering collaboration.

Unlocking Decentralized AI Training at Scale

DiLoCo's reduced synchronization frequency, on-chain management, and proof-of-learning-based validation tackle bandwidth constraints, centralization risks, and trust issues. However, to enable decentralized training of truly large models, we must also solve the scalability problem. As models grow to hundreds of billions of parameters, we cannot reasonably expect all participants to train the full model on their servers, this would be too harsh a constraint and would prevent many potential participants from joining the training. For now, we still expect participants to be able to store the full model on one server for inference purposes to maintain performance,

To address this challenge, various sharding approaches have been developed that divide models into distinct portions distributed across multiple nodes. This allows each participant to store and train only a subset of the model's parameters. This concept has been explored in several notable works. GShard [22] introduced efficient scaling techniques for trillion-parameter models using conditional computation and model parallelism in distributed Mixture-of-Experts (MoE) training; Decentralized Mixture-of-Experts [23] proposed decentralized training methods for MoE architectures, introducing a framework for training large models using unreliable consumer-grade hardware; and DiPaCo [24] presented techniques for distributed parameter coordination in large-scale training. Paths of shared 150M-parameter modules are trained using DiLoCo's synchronization (outer-gradient aggregation every 100 steps), achieving 45% faster training than dense baselines. The resulting model combines 256 paths that share some of the parameters, and has a total size of 20B parameters, requiring to store only 150M on each node during training.

Our framework allows combining the DiLoCo with these sharding approaches. Thus the maximum feasible model size grows significantly because each node handles only a portion of the model. During training, nodes maintain only the parameters for their assigned portions and share updates related to these components when synchronization events occur. With this combined approach, training models at large scales becomes viable, including models comparable to DeepSeek's R1 with its 671B parameters and larger as the computational and memory burden is distributed across the network rather than concentrated on individual nodes.

By integrating efficient synchronization mechanisms, on-chain management, validation, and model sharding techniques, we create a comprehensive framework for decentralized, large-scale AI training. This approach opens possibilities for community-driven development of sophisticated AI systems through collaborative, distributed efforts.